ESTRUCTURA DE DATOS

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

ESTRUCTURA DE DATOS
Programas de Educación a Distancia
Universidad de Cartagena
2017; [N°] Pág.; 21.5 X 27.9 cm

© Bubok Publishing S.L., 2017
1ª edición
ISBN:
Impreso en Colombia
Impreso por…

ESTRUCTURAS DE DATOS

No está permitida la reproducción total o parcial de este libro o por cualquier medio o método de éste sin previa autorización de **la Universidad de Cartagena**, del Autor(a) y la **Empresa Editorial**. Ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sean electrónicos, mecánicos, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del Copyright.

DERECHOS RESERVADOS © 2017, respecto a la primera edición en español, por

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

CONTENIDO

Pág.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

UNIDAD III.

Estructuras de Datos Dinámicos (No Lineales)

3. Unidad 3: ESTRUCTURAS DE DATOS DINÁMICOS (NO LINEALES) – PARTE 2

3.1. Presentación de la unidad.

3. ÁRBOLES

Aplicando una estructura de árbol, el estudiante realizará un programa que represente su propio árbol genealógico, permitiendo consultar la información de sus padres, abuelos, bisabuelos y tatarabuelos y satisfaciendo las necesidades de búsqueda de cualquier integrante de la familia.

Al terminar este Módulo, el estudiante desarrollará, por medio de árboles B, un programa de rastreo de pedidos para una empresa de mensajería y paquetería. El programa debe satisfacer los requisitos de búsqueda de los clientes y almacenar la información de hasta 20,000 envíos. Lo anterior es básico en búsquedas eficientes y, además, optimiza el uso de memoria principal cuando se utilizan grandes cantidades de información.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

SABER CONCEPTUAL : (Teorías, Definiciones)
3.1. Árboles. 3.1.1. Terminología 3.1.2. Árboles binarios y su representación gráfica 3.1.3. Recorridos de árboles binarios 3.1.4. Representación de árboles en lenguaje C 3.1.5. Operaciones básicas en árboles 3.1.6. Árboles multicaminos 3.1.7. Árboles B 3.1.8. Inserción en árboles B 3.1.9. Ejercicios Resueltos 3.1.10. Aplicaciones
SABER HACER (Aplicaciones)
❑ Diferenciar las clases de Árboles, entre Binarios, AVL y B. ❑ Desarrollar los algoritmos de recorrido de Árboles. ❑ Realizar Búsqueda de elementos en un Árbol. ❑ Implementar los algoritmos más utilizados sobre árboles en aplicaciones concretas. ❑ Conocer las diversas aplicaciones computacionales de árboles
SABER SER (Valores)
Solidaridad
Amor por el saber
Actitud crítica.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

Cultura por el trabajo individual y en equipo.
Creatividad e imaginación.
Adaptación al cambio de trabajo en otro idioma.
Cumplimiento de las consultas y ejercicio.
Capacidad de razonamiento e interpretación
Responsabilidad y Autonomía
Constancia para llegar al resultado.
EVALUACION DEL APRENDIZAJE
EVIDENCIA DE CONOCIMIENTO (Evidencia oral o escrita)
• Trabajos en grupo y Exposición • Laboratorios • Examen escrito
EVIDENCIA DE DESEMPEÑO (Evidencia del hacer)
Resolución de Ejercicios Protocolos
EVIDENCIA DE PRODUCTO (Evidencia Tangible)
Trabajo de Investigación Proyecto de clase

3.2. Mapa Conceptual.

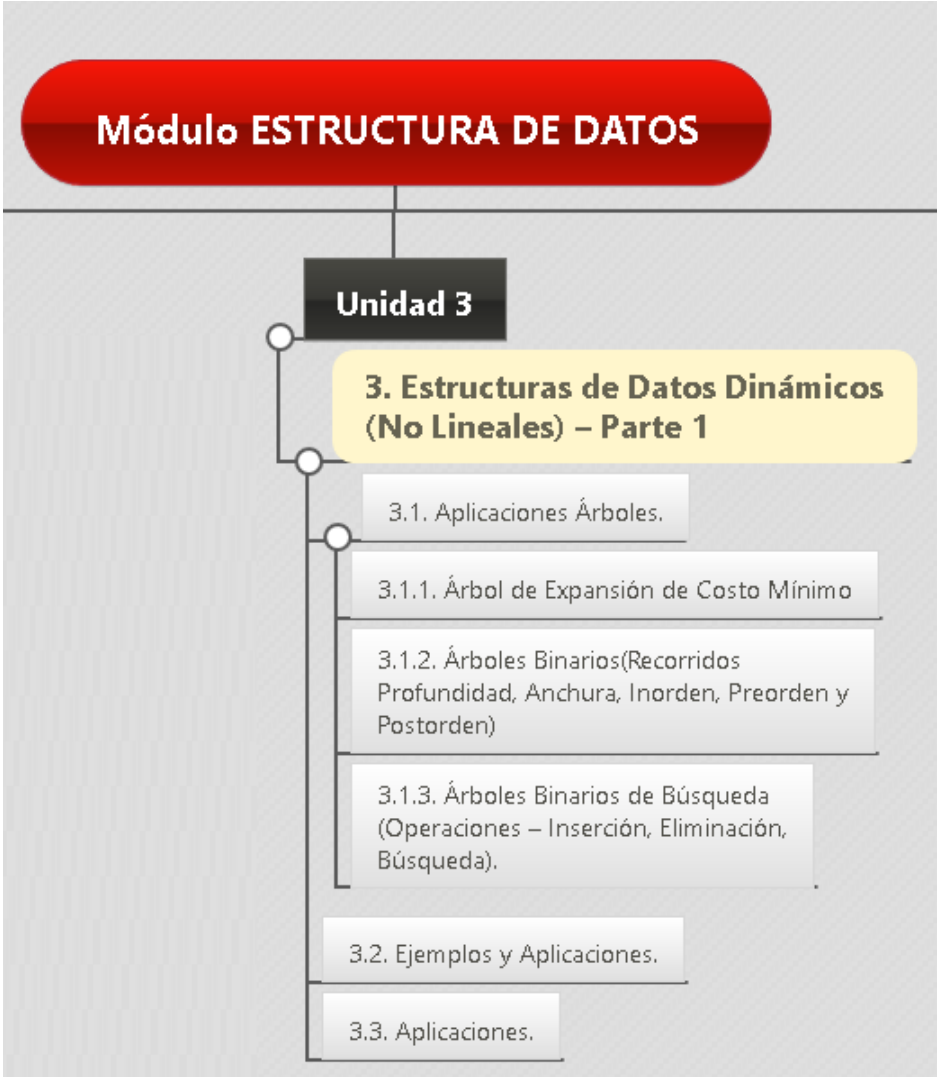


Ilustración 1. Mapa Conceptual Módulo ESTRUCTURA DE DATOS[1]

Ver mapa

[1] Fuente: Autoría Propia. Ref. Web: https://www.mindmeister.com/es/580337973/m-dulo-estructura-de-datos

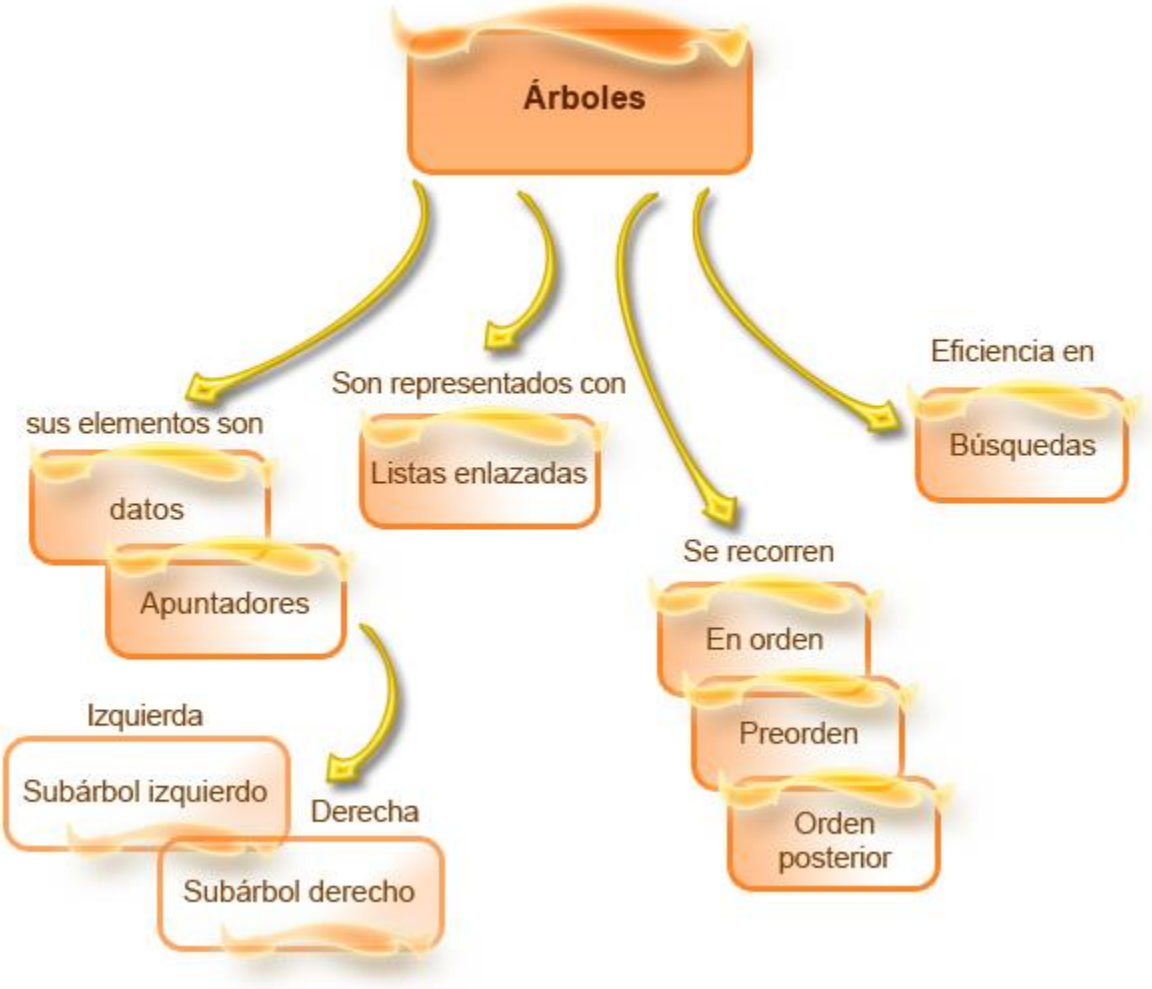
Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

3.3. Evaluación de Pre saberes.

Vamos a iniciar con unas preguntas, tal vez no las puedas responder en este momento, pero conforme avances en el Módulo las contestarás con facilidad.

1. ¿Sabías que la recursividad es la forma más sencilla de codificar los árboles?
2. ¿Por qué los árboles son la estructura de datos más eficiente para realizar búsquedas?
3. ¿Cuál es la principal ventaja de los árboles B?
4. ¿Sabías que los árboles B+ son usados en el acceso “índices” en una base de datos?
5. ¿Sabías que utilizando árboles B podemos hacer que sólo parte del árbol se encuentre en memoria principal y el resto en un almacenamiento secundario?

3.4. Lección 1: Árboles Binarios



3.4.1. Introducción

Cuando hablamos de árboles siempre pensamos en uno grande que nos pueda proteger del sol y, ¿por qué no?, recordamos cuando éramos niños y jugábamos competencias para ver quién lo trepaba más rápido; además, siempre nos imaginamos un árbol compuesto de una raíz, ramas y hojas. Este concepto no está tan alejado de las estructuras de datos: de ahí el nombre de árbol.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

Los árboles representan una de las estructuras de datos más potentes, sobre todo en la realización de búsquedas, ya que en un árbol los elementos pueden ser ordenados desde que se van ingresando.

Los árboles son las estructuras no lineales y dinámicas de datos más importantes en computación, razón por la cual dedicaremos dos Módulos para analizarlas.

Podríamos seguir hablando de las bondades de los árboles, pero mejor lo dejamos para cada uno de los temas.

3.4.2. Terminología

Árbol: Es una estructura jerárquica que consiste en una colección de elementos u objetos llamados nodos, en donde uno de estos nodos es llamado raíz. En un árbol se crea una serie de conceptos como son:

Raíz: Es un nodo que no tiene padre (el nodo 10).

Nodo hoja: Son nodos que no tienen hijos (nodos 15, 14, 40).

Rama: Es aquella que tiene hijos pero que no es el nodo raíz (nodos 5, 8, 23).

Orden: Es el número potencial de hijos que puede tener cada elemento de árbol. Si un árbol sólo puede tener dos hijos es de orden 2, si tiene tres hijos es de orden 3. En este módulo se estudiarán los árboles de orden 2, tal es el caso del ejemplo que se presenta.

Nivel: es el numero de arcos que se deben recorrer para llegar a un nodo determinado. Por definición la raíz tiene nivel 1.

Es la distancia que existe entre el nodo raíz y otro nodo. En el ejemplo: el nodo 10 tiene un nivel 0, el nodo 15 tiene un nivel 2, el nodo 40 tiene un nivel 3.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

Altura: Es el nivel del nodo de mayor nivel. En este ejemplo el nodo de mayor nivel es el 40, por lo que la altura del árbol es 3.

Grado: el numero de subárboles de un nodo o numero de descendientes directos de un nodo.

Un nodo de grado 0: se llama nodo terminal.

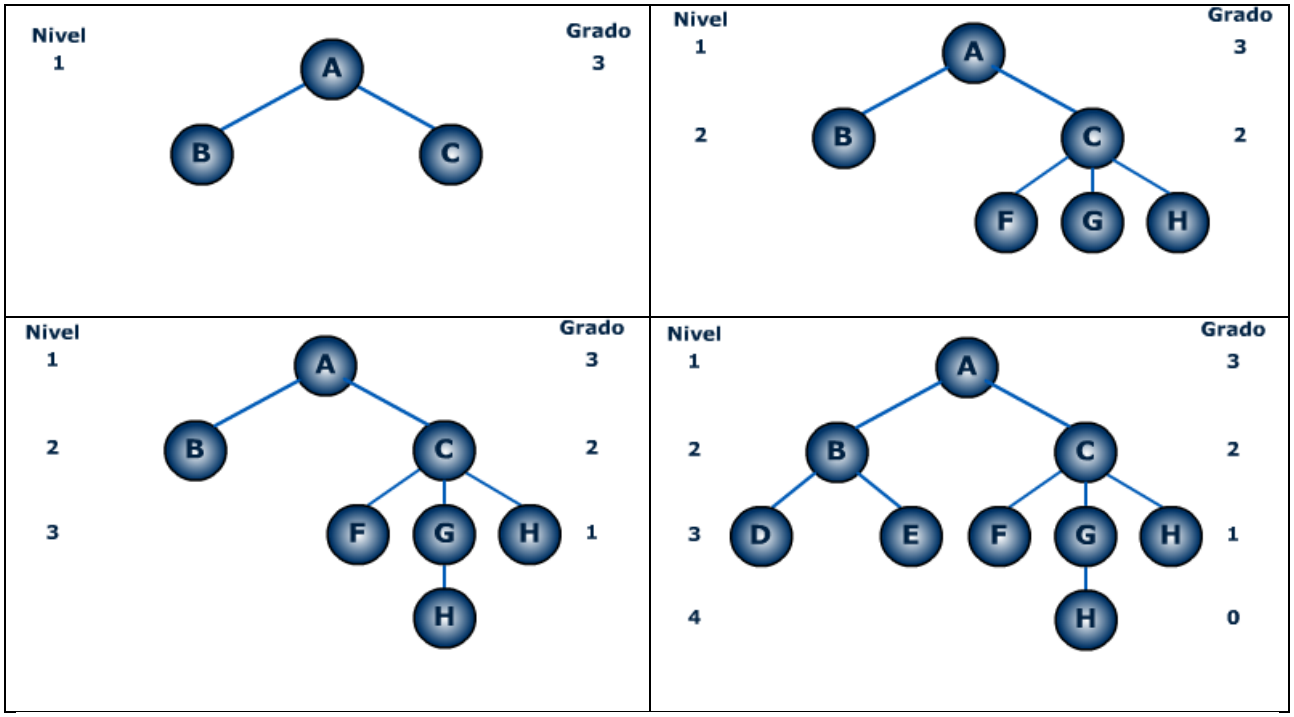


Figura 1 Niveles y grados en un árbol.

3.4.3. Árboles

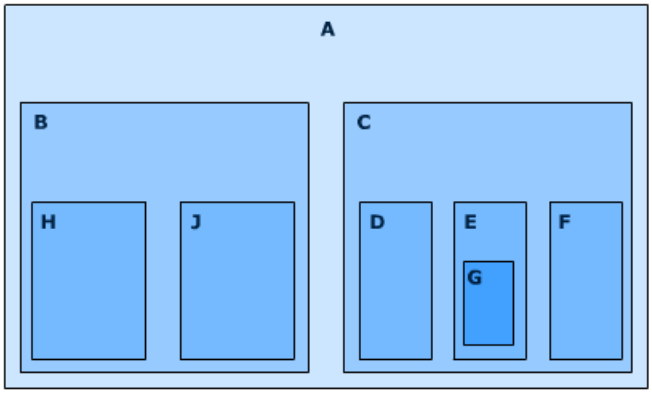
Son las estructuras no lineales más importantes que se emplean en muchas aplicaciones de computación.

La estructura de un árbol significa una relación de ramificación entre nodos, lo cual implícitamente conlleva una relación jerárquica.

Definición. Un árbol es un conjunto finito de nodos T tales que:

1. Hay un nodo especial llamado raíz
2. Los otros nodos se particionan en M conjuntos T_1, T_2, ..., T_M. Los conjuntos T1, T2, ..., TM son árboles y se llaman subárboles de T.

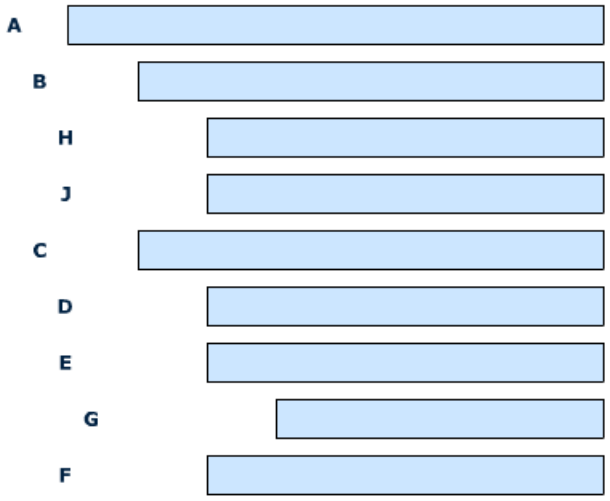
- **Diagramas de Venn**



- **Por expresiones**

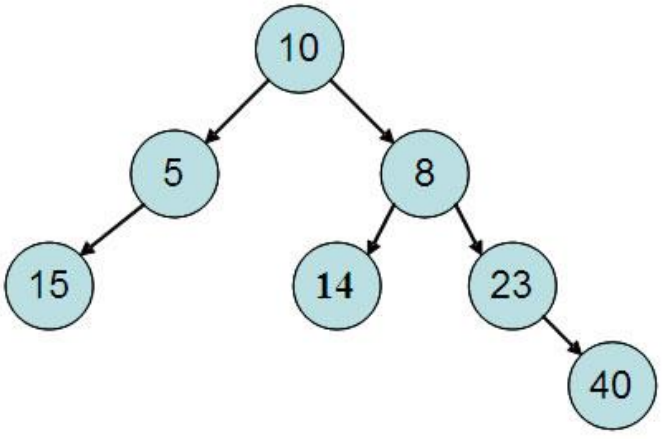
(a(b(h, j),c(d, e(g),f))

- **Por indentacion**



3.4.4. Árboles binarios y su representación gráfica

Como se visualiza en la figura, el árbol está representado por un nodo llamado raíz, con dos apuntadores, uno a la izquierda y otro a la derecha, y a su vez el nodo de la izquierda también puede tener apuntadores a la izquierda y a la derecha (es decir, cada nodo puede ser un subárbol).



Un árbol binario t es un conjunto finito de nodos tales que

1. t es vacío
2. t consiste de un nodo r, llamado raíz de t y dos árboles disyuntos t_1 y t_2 llamados subárboles izquierdo y subárbol derecho.

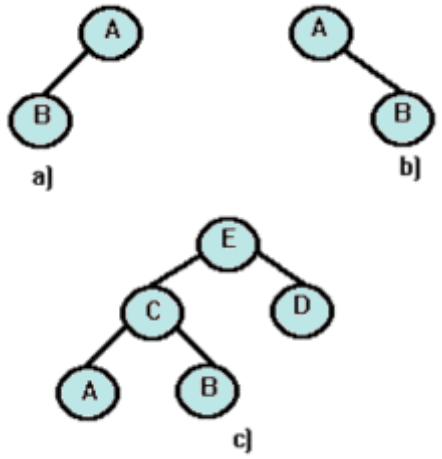


Figura 2. Árboles binarios) subárbol derecho vacío. b) subárbol izquierdo vacío. c) un árbol binario.

Dos árboles son distintos cuando sus estructuras son diferentes. en la figura 6.2. todos los árboles son distintos.

Dos árboles son similares cuando sus estructuras son idénticas pero la información de los nodos es diferente.

Dos árboles son equivalentes cuando son similares y además tienen la misma información.

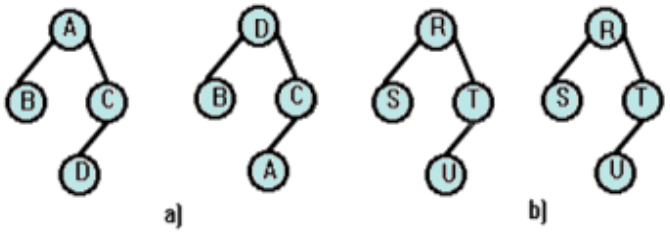


Figura 3. a) árboles distintos. b) árboles equivalentes

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

3.4.5. Recorrido de árboles binarios

Una operación común en los árboles es recorrerlos, es decir, visitar cada uno de los nodos. Para realizar esta operación existen tres formas:

En orden previo:

- Visitar la raíz.
- Recorrer el subárbol izquierdo en orden previo.
- Recorrer el subárbol derecho en orden previo.

En orden:

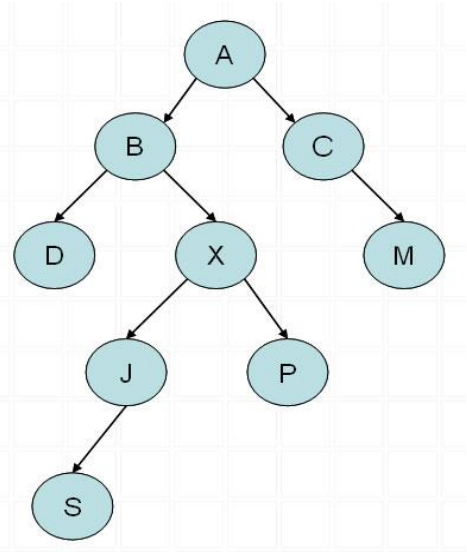
- Recorrer el subárbol izquierdo en orden.
- Visitar la raíz.
- Recorrer el subárbol derecho en orden.

En orden posterior:

- Recorrer el subárbol izquierdo en orden posterior.
- Recorrer el subárbol derecho en orden posterior.
- Visitar la raíz.

Ejemplo: Vamos a realizar el recorrido de este árbol de las tres formas:

Orden previo	A-B-D-X-J-S-P-C-M
Orden	D-B-S-J-X-P-A-C-M
Orden posterior	D-S-J-P-X-B-M-C-A



3.4.6. Conversión de Árboles No Binarios a Binarios

Debido a que los árboles binarios son estructuras muy estudiadas, por esta razón en muchas aplicaciones se convierten árboles no binarios en árboles binarios. para realizar la conversión se siguen las siguientes reglas:

1. Los nodos hermanos se enlazan en forma horizontal.
2. El nodo raíz se enlaza en forma vertical con el subárbol izquierdo.
3. Se gira la estructura resultante 450.

Ejemplo:

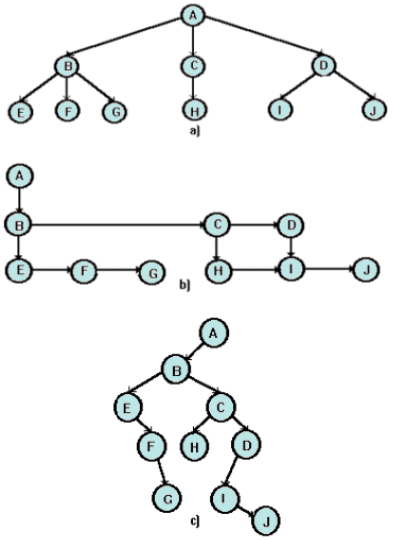
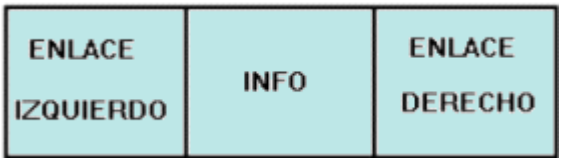
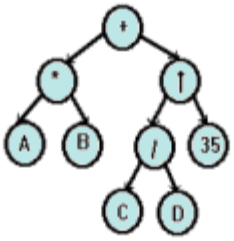


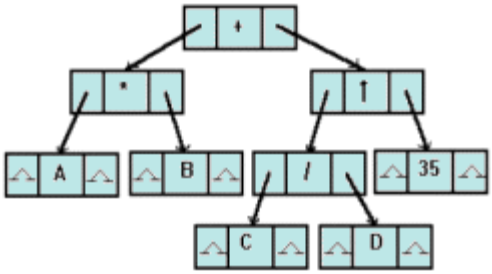
Figura 4. Conversión de un árbol no binario a binario.
a) árbol inicial. b) árbol binario después de aplicar las reglas 1 y 2.
c) árbol equivalente después del paso 3.

Los árboles se pueden representar en memoria estática a través de estructuras arreglo. también se puede hacer a través de estructuras dinámicas. aquí se trabajara el manejo dinámico, el nodo de un árbol es:



Ejemplo: Representar la expresión (A*B)+(C/D)^ 35





Una operación muy importante es el recorrido de los árboles binarios. recorrer un árbol significa visitar los nodos del árbol en forma sistemática, de esta manera se garantiza que se visitan todos los nodos una sola vez.

Hay tres métodos, recursivos, para hacer el recorrido de un árbol:

- Preorden: raíz, subárbol izquierdo, subárbol derecho.
- Inorden: subárbol izquierdo, raíz, subárbol derecho.
- Posorden: subárbol izquierdo, subárbol derecho, raíz.

Ejemplo:

Preorden: abdhecfgijk.

Inorden: dhbeafcigkj.

Posorden: hdebfikjgca.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

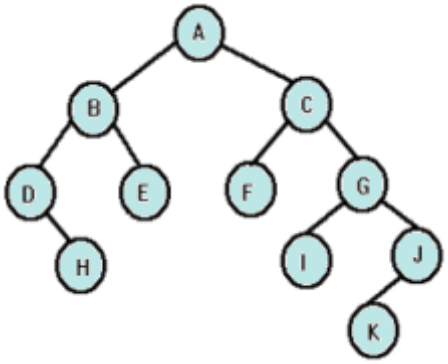


Figura 5. árbol binario del ejemplo para hacer los recorridos

3.4.7. preorden (t)

```
comience
crea_pila(tope)
p<=t
mientras ((tope <> ) v (p<> )) ejecute
si ( p <> )entonces
imprima info (*p)
inser_pila (tope,p)
p <= ei (*p)
sino
elim_pila (tope,p)
p <= de (*p)
fin-si
fin-mientras
termine
```

3.4.8. inorden (t)

```
comience
crea_pila (tope)
```


Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

```
p <= t
mientras (tope <.> ) v ( p <> )ejecute
si ( p<> )entonces
inser_pila (tope,p)
p <= el (*p)
sino
elim_pila (tope,p)
imprima info (*p)
p <= ed(*p)
fin-si
fin-mientras
termine
```

3.4.9. posorden (t)

```
comience
crea_pila(tope)
p<=t
mientras (p<> ) ejecute
si (p<> ) entonces
inserte_pila (tope,p,o)
p<= ei (*p)
sino
elimin_pila (tope,q,k)
p<=ed (*q)
si (( p = ) v (k=1) entonces
imprima info (*q)
```

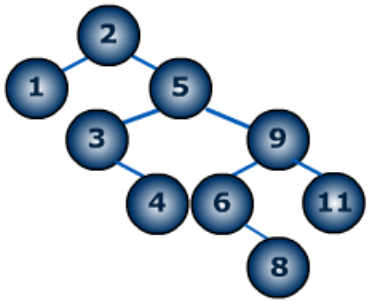
```
sino
inserte_pila (tope,q,1)
fin-si
fin-si
fin-mientras
termine
```

Pero para ser el recorrido de un árbol primero se debe conformar o llenar. Naturalmente las inserciones en un árbol depende de la aplicación.

3.4.10. Ordenamiento AVL

Para todo nodo los valores de los nodos del subárbol izquierdo de t deben ser menores o iguales al valor de t. así mismo todos los valores del subárbol derecho de t son mayores o iguales al valor de t.

Ejemplo: Se tiene la lista
2,5,9,6,1,3,4,8,11

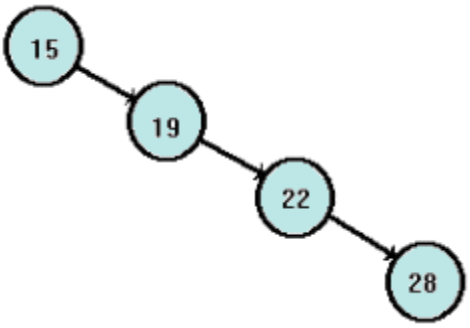


Una vez se ha hecho la inserción se hace recorrido en inorden.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

inser_avl (a,i)
comience
crea_arbol(t)
k<=1
asigna_nodo (p,a[k ,i); t <=p; k <=2
mientras (k<=n), ,(i=o)) ejecute
s<=t
mientras (s <>)ejecute
r<=s
si (a[k >*´s.info) entonces
s<=ed(*s)
sino
s<= ei (*s)
fin-si
fin-mientras
si(datos>info(*r))entonces
de (*r)<=p
sino
ei(*r)<=p
fin-si
k<= k+1
fin-mientras
termine

Este tipo de ordenamiento permite realizar eficientemente operaciones de búsqueda. sin embargo, si el árbol crece descontroladamente, el rendimiento puede disminuir; por ejemplo:



3.4.11. Ordenamiento Floyd

El árbol balanceado es un árbol balanceado cuya diferencia de niveles entre ramas no es mayor de uno.

Al insertar un elemento en un árbol balanceado deben distinguirse los siguientes casos:

1. Las ramas izquierdas (ir) y derecha (rd) del árbol tienen la misma altura (hri=hrd), por lo tanto:
   - 1.1. si se inserta un elemento en ri entonces hri será mayor a hrd.
   - 1.2. si se inserta un elemento en rd entonces hrd será mayor a hri.

   Obsérvese que en cualquiera de los dos casos mencionados (1.1 y 1.2), no se viola el criterio de equilibrio del árbol.
2. Las ramas izquierda (ri) y derecha (rd) del árbol tienen altura diferente (hri <> hrd):
   - Supóngase que hri<hrd:
     - Si se inserta un elemento en ri entonces hri será igual a hrd. {las ramas tienen la misma altura, por lo que se mejora el equilibrio del árbol}.
     - Si se inserta un elemento en rd entonces se rompe el criterio de equilibrio del árbol y s necesario restructurarlo.
   - Supóngase que hri > hrd:
     - Si se inserta un elemento en ri, entonces se rompe el criterio de equilibrio del árbol y es necesario restructurarlo.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

- Si se inserta un elemento en rd, entonces hrd será igual a hri. {las ramas tienen la misma altura, por lo que se mejora el equilibrio del árbol.

3.4.12. Recorridos en Forma Recursiva

3.4.12.1. preorden (t)

```
comience
si (t <> )entonces
imprima          info          (*t)
preorden                       (ei(*t))
preorden (ed(*t))
sino
fin-si
termine
```

3.4.12.2. inorden (t)

```
comience
si (t <> ) entonces
inorden (ei(*t))
imprima info (*t)
inorden (ed(*t))
sino
fin-si
termine
```


Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

3.4.12.3. posorden(t)

```
si(t<> ) ejecute
posorden(ei(*t))
posorden (ed(*t))
imprima info (*t)
sino
fin-si
termine
```

3.4.13. Operaciones básicas en árboles

Las operaciones en árboles son casi las mismas que para las listas:

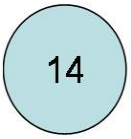
- Añadir elementos:
  - Recorrer el árbol completo.
  - Borrar árbol.
  - Búsqueda de un elemento

Añadir elementos.

Por lo regular los elementos en un árbol se insertan de tal forma que queden ordenados. En esta sección analizaremos cómo insertar elementos utilizando el ***recorrido*** en ***orden***. Suponemos que se van a insertar los siguientes elementos:

14-16-2-17-15

Inserción del nodo 14: Como no existen elementos en el árbol, el ***nodo 14***será la raíz.

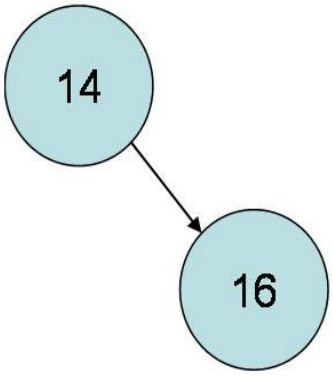


Inserción del nodo 16: Para insertar este nodo recorremos el árbol en ***orden*** y además buscamos que el elemento quede ordenado. Es decir:

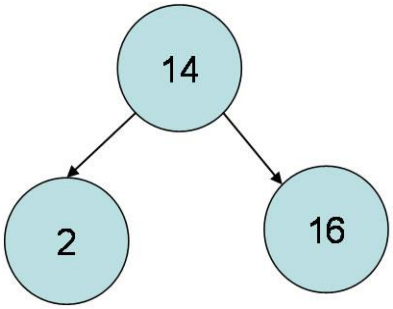
Nodo a insertar menor que el padre: Lo insertamos a la izquierda.

Nodo a insertar mayor que el padre: Lo insertamos a la derecha.

En este ejemplo el ***nodo 16*** es mayor que el padre por lo tanto lo insertamos a la derecha

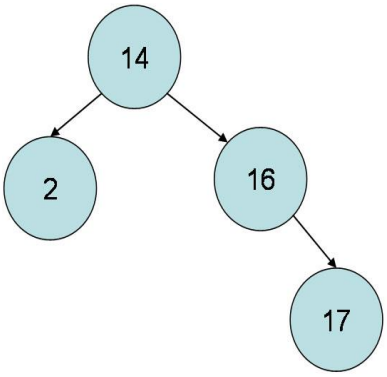


Inserción del nodo 2: El elemento 2 es menor que el padre por lo tanto es insertado a la izquierda.



Inserción del nodo 17: Como recordaremos el recorrido del árbol se hace en***orden*** por lo tanto:

El nodo es mayor que la raíz y se insertará a la derecha
Encontramos que a la derecha ya existe el ***nodo 16*** y que el ***nodo 17***es mayor, por lo tanto, hacemos el mismo procedimiento que en el paso 1 es decir, insertamos el elemento a derecha del ***nodo*** 16.

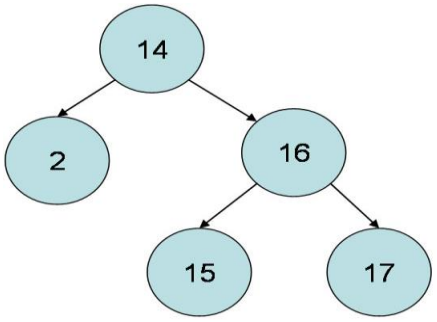


Inserción del nodo 15:

1. El ***nodo 15*** es mayor que la raíz, por lo que la inserción se realizará por la derecha.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

2. Llegamos al ***nodo 16*** y de nuevo verificamos si es mayor o menor que el nodo a insertar, como se detecta que es menor se inserta a la izquierda.



Tal vez te preguntarás:

¿Por qué complicarse tanto al insertar los elementos en el árbol? Podríamos insertarlos conforme vayan llegando sin seguir un recorrido.

La respuesta es sencilla: el insertarlos de esta forma, permite ordenar los elementos del árbol al recorrerlo. Por lo que al hacer un ***recorrido en orden*** obtendremos los elementos ordenados.

2-14-15-16-17

Comentarios: Como habrás detectado para insertar se hacen tareas repetitivas hasta encontrar el lugar adecuado para el nodo, para representar inserción de árboles en un lenguaje de programación se puede usar un código iterativo hasta llegar a un nodo desocupado pero este código se vuelve muy complicado, una forma sencilla de insertar los elementos es hacer uso de un algoritmo recursivo.

Como ya hemos analizado la forma en la que se añaden elementos en un árbol, ahora sí podemos desarrollar el código en lenguaje C para insertar elementos en un árbol. Este algoritmo será recursivo, debido a que es más sencillo de codificar que el iterativo.

Inserción de nodos en lenguaje C: Al insertar los elementos se usa una función recursiva con dos parámetros (un apuntador a la ***raíz*** y el valor a insertar). El algoritmo es:

1. Si ***raíz*** es nulo

a. Crear el nuevo nodo;
b. En el campo información asignar el ***valor*** que recibimos como parámetro;
c. Los ***apuntadores a izquierda y derecha*** se inicializan con nulo;
d. ***raíz=nuevo***.

2. Si el apuntador no es nulo

a. Si la **raíz** en su campo ***información*** es mayor al ***valor*** que se recibió como parámetro (indica que el elemento es menor; por lo tanto, se insertará por la izquierda);

i. Llamar a la misma función de forma recursiva con los parámetros ***raíz ->izquierda***;

b. Si la **raíz** en su campo ***información*** es menor al ***valor*** que se recibió como parámetro (indica que el elemento es mayor; por lo tanto, se insertará por la derecha);

i. Llamar a la misma función de forma recursiva con los parámetros ***raíz ->derecha;***

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

```
void inserta(arbol_binario*&raiz,intvalor)
{
      arbol_binario*nuevo;
       if(!raiz)
 {
                        //creamos el nodo
                        nuevo=(arbol_binario*)malloc(sizeof(ar
bol_binario));
                        nuevo->info=valor;
                        nuevo->izq=NULL;
                        nuevo->der=NULL;
                        raiz=nuevo//para poner elnuevo nodo
en su lugar
              }
              else
              {
                        if(raiz->info>=valor)
                                  inserta(raiz->izq, valor);//el
nuevonodo se insertará por la izquierda
                        else
                                  inserta(raiz->der, valor);//el
nuevonodo se insertará por la derecha
              }
              return;
}
```

Recorrido del árbol:

Como ya vimos en el tema anterior, los árboles pueden ser recorridos en orden previo, en orden, y en orden posterior. El código para realizar cada una de estas operaciones es:

Recorrido en orden previo

```
void orden_previo(arbol_binario*raiz)
{
        if(raiz)//sitodaviaexisten nodos hacia abajo
        {
                printf("%d",raiz->info);
                orden_previo(raiz->izq);
                orden_previo(raiz->der);
        }
        return0;
}
```

Recorrido en orden

```
void orden(arbol_binario*raiz)
{
        if(raiz)//sitodaviaexisten nodos hacia abajo
        {
                orden(raiz->izq);
printf("%d",raiz->info);
                orden(raiz->der);
        }
        return0;
```

```
}
```

Recorrido en orden posterior

```
void orden_posterior(arbol_binario*raiz)
{
        if(raiz)//sitodaviaexisten
nodos hacia abajo
        {

printf("%d",raiz->info);
                orden_posterior(raiz-
>der);
orden_posterior(raiz->izq);
        }
        return0;
}
```

Borrar árbol:

Así como un árbol puede ser creado, también es posible eliminar cada uno de los nodos que lo componen.

```
void
destruye(arbol_binario*&raiz)
{
        if(raiz)
        {
                destruye(raiz-
```


Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

```
>izq);
                    destruye(raiz-
>der);
                    free(raiz);
                    raiz=NULL;
                }
}
```

Buscar un elemento:

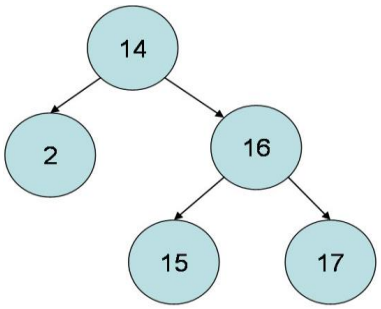
Una de las ventajas que tienen los árboles respecto de otras estructuras como listas, pila o colas, es la eficiente búsqueda de elementos, debido a que los elementos en un árbol se pueden insertar ordenados. En los arreglos las búsquedas se pueden hacer eficientemente si el arreglo se encuentra ordenado, pero la inserción y eliminación resultan costosas. Por el contrario, en las listas, dichas operaciones se pueden realizar fácilmente, pero la búsqueda no es eficiente, porque incluso en una búsqueda nos puede llevar recorrer todo el árbol (si el elemento buscado se encuentra en último lugar).

Cuando buscamos siempre tenemos tres opciones:

Opción a) Buscar por la izquierda (Si el elemento a buscar es < que el nodo padre);
Opción b) Buscar por la derecha (Si el elemento a buscar es > que el nodo padre);
Opción c) Mostrar los datos del nodo (Si el elemento a buscar es igual al nodo padre).

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

Ejemplo: Suponemos que deseamos buscar el elemento **17** en el árbol:



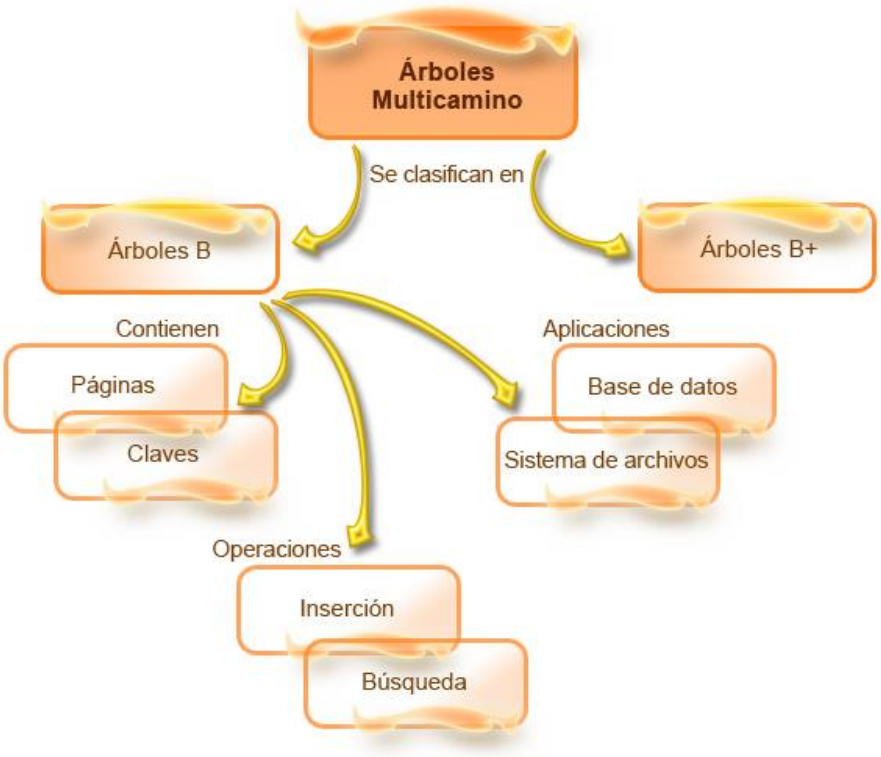
1. Como el 17 es mayor que el nodo padre (14) buscaremos por la derecha;
2. Llegamos al nodo padre 16 y como el 17 es mayor de nuevo buscaremos por la derecha;
3. Ahora el 17 es el nodo padre: ¡¡lo hemos encontrado!!

```
void busqueda(arbol_binario *&raiz, int dato)
{
        if(dato < raiz->info)
        {
                if(raiz->izq == 0)
                        printf("El nodo no se
encuentra en el arbol");
                else
                        busqueda(raiz->izq,
dato);
        }
        else
        {
                if(dato > raiz->info)
                {
```


Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

```
                                    if(raiz->der ==NULL)
                                            printf("El nodo
no se encuentra ");
                                    else
                                            busqueda(raiz-
>der, dato);
                            }
                            else
                            {
                                    if(raiz->info ==dato)
                                    {
                                            printf("DATO
ENCONTRADO ES %d ", raiz->info);
                                            return;
                                    }
                                    else
                                            printf("El nodo
no se encuentra ");
                            }
                    }
}
```

3.5. Lección 2: Árboles B



3.5.1. Introducción

Ya casi terminamos el curso, sólo nos restan dos Módulos por estudiar; si bien a simple vista parecen difíciles, conforme vayas avanzando verás que en realidad no lo son. Ahora bien, en este módulo veremos una variante de los árboles binarios estudiados en el módulo pasado, nos referimos a los árboles B, en donde cada nodo puede tener más de dos hijos; además, estos árboles crecen de abajo hacia arriba.

Este módulo está compuesto por:

- **Árboles multicamino:** Son la base de los árboles B y B+, este tema no será analizado a profundidad sólo se estudia porque de este tema surgieron los árboles B.
- **Árboles B:** Es el tema principal de este módulo, te recomiendo leer cuidadosamente la definición de árbol B para que tu desempeño sea un éxito.
- **Inserciones en árboles B:** La inserción en árboles B parece un poco complicada, pero leyendo el algoritmo detalladamente verás que es muy simple.
- **Búsqueda en árboles B:** Las búsquedas en árboles B se basan en ir comparando en las páginas del árbol en el lugar donde se encuentra la clave.

Ya que hemos visto a grandes rasgos los temas de este módulo, ¡iniciemos!

3.5.2. Árboles multicamino

- Los **árboles multicamino** son estructuras de datos de tipo árbol, en donde cada nodo tiene orden mayor a dos.
- Los árboles multicamino son una variante de los árboles binarios, analizados en el módulo anterior.

¿Cuál es la principal ventaja de los árboles multicamino?

La principal ventaja de este tipo de árboles consiste en que existen más nodos en un mismo nivel que en los árboles binarios con lo que se consigue que, si el árbol es de búsqueda, los accesos a los nodos sean más rápidos.

Aplicaciones de árboles multicamino

Existen muchas aplicaciones en las que el volumen de la información es tal, que los datos no caben en la memoria principal y es necesario almacenarlos, organizados en archivos, en dispositivos de almacenamiento secundario. La solución a este problema es la utilización de árboles multicamino, ya que permiten recuperar los datos de los dispositivos en forma eficiente. Por lo anteriormente mencionado, los árboles multicamino son muy usados en bases de datos y sistemas de archivos.

Clasificación de árboles multicamino

Árboles B: Son árboles de búsqueda. En donde cada nodo (página) de orden ***d*** debe tener ***2d*** claves como máximo y ***d*** claves como mínimo. Estos árboles serán los que analicemos a detalle en este módulo.

Árboles B+: Es una variante de los árboles B, su principal característica radica en que todas las claves se encuentran en las hojas y, por lo tanto, cualquier camino desde la raíz hasta alguna de las claves tiene la misma longitud.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

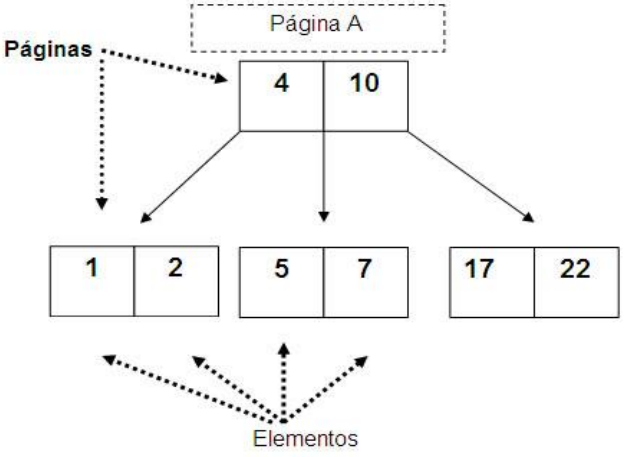
3.5.3. Árboles B

Fueron desarrollados por Rudolf Bayer y Eduard M. McCreigh.
La letra B no significa "binario", ya que:

a. Los árboles B nunca son binarios.
b. Y tampoco es porque sean árboles de búsqueda, ya que en inglés se denominan B tree.
c. Tampoco es porque sean balanceados, ya que no suelen serlo.

Antes de iniciar el tema a profundidad vamos a analizar algunos conceptos y su nomenclatura, que si bien ya fueron estudiados en el módulo anterior, algunos varían.

- **Página:** Es una variante del concepto de *nodo* utilizado en el módulo anterior, la diferencia es que contiene más de un elemento dentro del mismo nodo.
- **Clave o elemento (m):** Se refiere a cada componente de la página.
- **Orden o grado (d):** El número de hijos que puede tener una página.



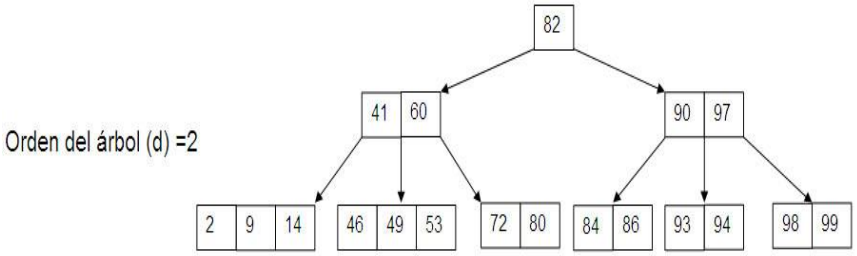
Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

En la figura el orden del árbol es 3: porque la página A tiene 3 hijos.

De acuerdo con Cairo (2002) un árbol se define formalmente de la siguiente manera:

1. Cada página, excepto la raíz, contiene entre d y 2d elementos (recuerda que d representa el grado del árbol).
2. Cada página, excepto la página raíz y las páginas hojas, tiene entre d + 1 y 2d + 1 descendientes. Se utilizará *m* para expresar el número de elementos por página.
3. La página raíz tiene, por lo menos, dos elementos.
4. Las páginas hojas están todas al mismo nivel.

Analicemos el siguiente árbol para verificar si cumple con los requisitos para ser un árbol B.



1. **¿Cada página, excepto la raíz, contiene entre d y 2d elementos?**

d = 2

Efectivamente, en el ejemplo cada página contiene entre 2 a 4 elementos.

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia

2. **¿Cada página, excepto la página raíz y las páginas hojas, tiene entre d + 1 y 2d + 1 descendientes?**

$$d = 2$$

Cada página del árbol de la figura sí tiene entre 3 y 5 descendientes.

3. **¿La página raíz tiene, por lo menos, dos descendientes?**

La raíz sí tiene dos descendientes.

4. **¿Las páginas hojas están todas al mismo nivel?**

Sí, todas las hojas tiene el mismo nivel.

Como el árbol cumple con las definiciones establecidas, se puede decir que sí es un **árbol B.**

3.5.4. Inserción en árboles B

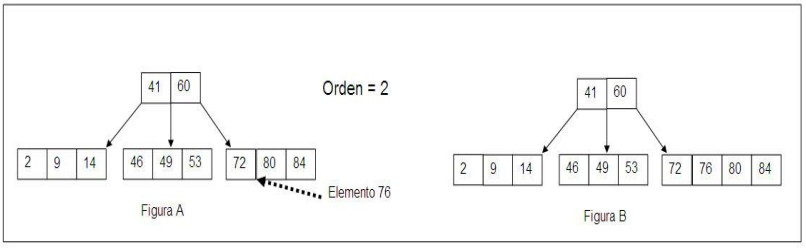
Como habrás visto en el módulo anterior, los árboles tienen una similitud, tal es el caso de la búsqueda, en donde los elementos menores se encuentra a la izquierda de la página y los mayores a la derecha. Al insertar elementos en un árbol B siempre debes tener presente que:

Todas las hojas tienen el mismo nivel: No importa cuántos elementos se inserten y si son menores o mayores a la raíz, todas las hojas deben tener el mismo nivel.

Crecen desde las hojas hasta la raíz: Los árboles B tiene una forma muy extraña de crecer porque crecen de abajo hacia arriba.
Al insertar elementos debes seguir las siguientes reglas:

1. Localizar la página donde corresponde insertar la clave.
2. Si el número de elementos de la página en donde se insertará el nuevo elemento es menor a 2d (m < 2d), se procede a insertar la clave.

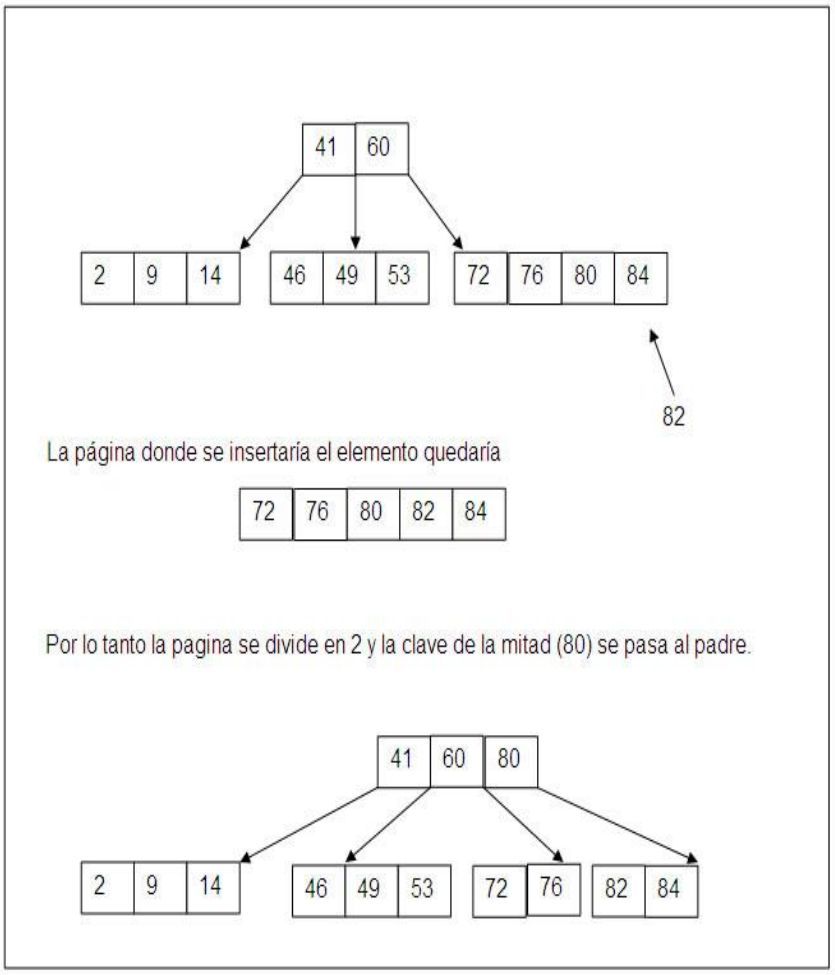
Ejemplo: Se desea insertar el elemento 76 en un árbol de orden 2.



3. Pero si la página se encuentra llena (m = 2d), la estructura del árbol cambia.

    a. Se divide la página en 2 y la clave del medio se sube a la página padre.

    Supongamos que, usando el árbol de la figura B se desea insertar la clave 82.


41 60
2 9 14
46 49 53
72 76 80 84
82
La página donde se insertaría el elemento quedaría
72 76 80 82 84
Por lo tanto la pagina se divide en 2 y la clave de la mitad (80) se pasa al padre.
41 60 80
2 9 14
46 49 53
72 76
82 84

REFERENCIAS BIBLIOGRÁFICAS

A continuación algunas *referencias bibliográficas* sobre este tema.

BIBLIOGRAFIA

FUENTES DOCUMENTALES

- INSUASTY R, Luis Delfín, Guía "A","B","C","D" de aprendizaje autónomo. Bogotá Colombia, Unad- Cafan
- MAURREN, Priestley . Técnicas y estrategias del pensamiento crítico. México D.F. 1996 (reimp .2000). Trillas.
- ARCEO B, Frida y Otro. Estrategias Decentes Para un Aprendizaje Significativo. Mexico D,F 1999. McGraw-HILL
- KENNETH C, louden . Lenguajes de programación (segunda edición). México D.F 2004. Thompson
- AGUILAR, Luis. Fundamentos de programación, algoritmos, estructura de datos y Objetos (tercera edición). España. 2003. McGRAW-HILL.
- AGUILAR, Luis. Programación en C++, Algoritmos, estructura de datos y Objetos España. 2000. McGRAW-HILL.
- DEYTEL Y DEYTEL. Como programar C++ (segunda Edición). México D.F. 1999. Prentice Hall. McGRAW-HILL
- FARREL, Joyce, introducción a la programación lógica y diseño. México D.F 2000. Thompson

Sitios Web

- http://www.geocities.com/david_ees/Algoritmia/curso.htm (Curso de algoritmia)
- http://www.ilustrados.com/publicaciones/EpZVVEZpyEdFpAKxjH.php (Lenguajes de Programación)
- http://www.ilustrados.com/buscar.php (Algoritmos)
- http://www.inf.utfsm.cl/~mcloud/iwi-131/diapositivas.html (Algoritmos)
- http://www.ucsm.edu.pe/rabarcaf/vonuep00.htm (Diccionario académico)
- http://www.funlam.edu.co/bired/index.asp-offset=0.htm (Aprendizaje Autónomo)

Facultad de Ingeniería - Programa de Ingeniería del Software Mod. a Distancia
Avenida del Consulado #Calle 30 No. 48 – 152,
Edificio Facultad de Ingeniería Tercer Piso
Teléfono: (575) 6752040, (5) 6752024 ext. 208 Fax: 6752040 – Apartado Aéreo 1382
www.unicartagena.edu.co - Cartagena de Indias D. T. y C. - Colombia